

1N-08-CR
181712

8P

FINAL REPORT

for

NASA Grant NAG-1-1066-FDP
Supplement-#1

Knowledge-Based Processing
for
Aircraft Flight Control.

For the Period Ending
October 17th, 1991

(Year-2 of a 3-year project)

Principal Investigator :
Dr. John H. Painter
Professor of Electrical Engineering
& Computer Science.

Department of Electrical Engineering
Texas A&M University
College Station, Texas 77843-3128
409, 845-7441
Internet : painter@zadok.tamu.edu

N94-13469

Unclas

G3/08 0181712

(NASA-CR-194074) KNOWLEDGE-BASED
PROCESSING FOR AIRCRAFT FLIGHT
CONTROL Final Report, period ending
17 Oct. 1991 (Texas A&M Univ.)
8 p

INTRODUCTION.

This is the final report for NASA Grant NAG1-1066 for the second year of effort.

The purpose of this grant is to develop algorithms and architectures for embedding artificial intelligence in aircraft guidance and control systems. With the approach adopted here, AI-computing is used to create an outer guidance loop for driving the usual aircraft autopilot. That is, a symbolic processor monitors the operation and performance of the aircraft. Then, based on rules and other stored knowledge, commands are automatically formulated for driving the autopilot so as to accomplish desired flight operations.

The focus is on developing a software system which can respond to linguistic instructions, input in a standard format, so as to formulate a sequence of simple commands to the autopilot. The instructions might be a fairly complex flight clearance, input either manually or by data-link. Emphasis is on a software system which responds much like a pilot would, employing not only precise computations, but, also, knowledge which is less precise, but more like "common-sense."

The approach is based on prior work to develop a generic "shell" architecture for an AI-processor, which may be tailored to many applications by describing the application in appropriate processor data bases (libraries). Such descriptions include numerical models of the aircraft and flight control system, as well as symbolic (linguistic) descriptions of flight operations, rules, and tactics.

FIRST YEAR PROGRESS.

During the first year, a top-level architecture was created, including the major software modules to be developed. These included the simulation of the aircraft and its autopilot, including an autoland system. The architecture is shown in Figure-1, below. This diagram shows explicitly the fact that there are three loops, being Inner, Middle, and Outer. Inner is the usual numerical flight control system and Outer is the pilot. The Middle Loop, being AI-based, symbolic processing, is now inserted between the two traditional loops.

The functionality of the Knowledge-based Processor is described as follows. The Interpreter analyzes numerical data from the aircraft sensors and interior variables from the numerical flight control system. Based on this data, the Interpreter identifies which one of a set of pre-defined flight conditions the aircraft is currently exhibiting. There is a pre-defined hierarchy of flight conditions, with most general flight operations being at the top of the hierarchy and most specific qualitative states at the bottom. Examples of the former are "take_off" and "climb_out." The latter include "stall," "accelerate," and "turning."

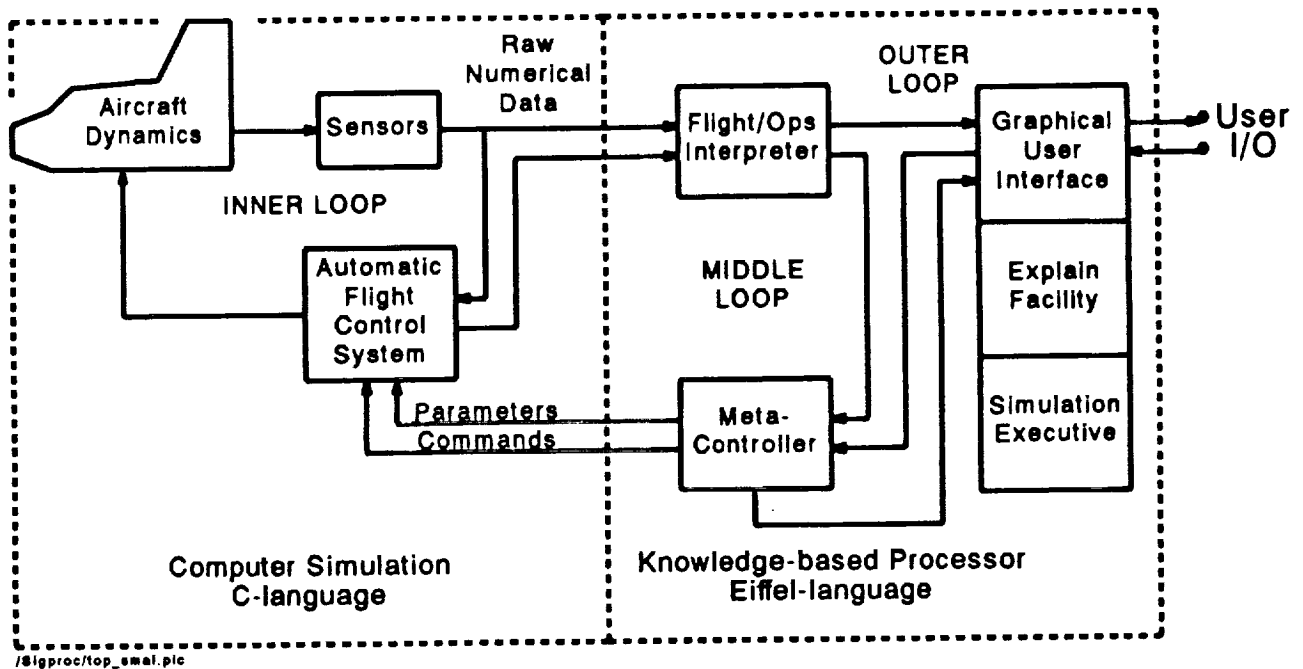


Figure 1. Top-level Data-flow Architecture.

Performances of interpreted flight operations and of the aircraft systems in that flight operations context are also evaluated. Linguistic descriptions of the interpreted operations and performance are communicated to the pilot through a Graphical User Interface. More detailed explanations are also available to the pilot on request.

The Meta-Controller functions to create inputs to the autopilot, to implement clearances and other standard format commands entered either directly by pilots or by some other means, such as data-links. The meta-controller is not just the "front-end" of a current state-of-the-art Flight Management System. Rather, it is an artificially intelligent system which avails itself of the past history (memory) of the Interpreter, in much the same way as would a human pilot. It recognizes constraints implied by the actual aircraft system performance, rather than just its ideal performance. It also avails itself of stored rules concerning techniques of flight, in order to yield a comfortable ride, rather than just one satisfying numerical constraints. Finally, the Meta-Controller is able to explain its formulation of autopilot inputs, upon request of the pilot.

During the first year, a choice of simulation model was made, in cooperation with the sponsor. This is a twin-jet Boeing-737 transport, in longitudinal axis only. The flight regime extends from high-altitude cruise through descent, approach to landing, and final landing attitude. A straight-in approach is assumed. The simulation employs five fundamental states, as per Roskam ^[1]. A piece-wise linear simulation, based on selected trim-points, was written during the summer of 1990 by Miss Emily Glass, during her residency at NASA LaRC. The simulation is in the language, C.

A software development environment was selected to support the development of the AI modules. An environment and language, named EIFFEL™ was chosen because it is created for object-oriented modeling and programming. An excellent textbook is available for EIFFEL ^[2]. Programs are written in EIFFEL and are debugged on-line. At run-time, the EIFFEL code is compiled into ANSI-standard C source code and then recompiled into machine executable code for the SUN workstation.

During the first year, an extensive investigation was made of many different theoretical research areas, to determine if a theoretical foundation for this work could be constructed from the diverse literature, thereof. These included Decision Science ^[3], Artificial Intelligence ^[4], Expert Systems ^[5], Knowledge-based (Qualitative) Reasoning ^[6], ^[7], Fuzzy Control ^[8], and Discrete-Event Dynamic System Control ^[9].

A theoretical basis was found in the literature for the separation of the symbolic processing guidance loop into the two blocks shown as the "Middle Loop of Figure-1, above. These were named Interpreter and Meta-Controller. The theoretical justification was found in the Decision Science literature ^[3]. However, it was later realized that this separation is a dual of a result well known in Stochastic Control, being the Separation Theorem ^[10]. Thereafter, other dualities were sought between "what works in AI," and "what works in Stochastic Decision, Estimation, and Control."

SECOND YEAR RESULTS :

The C-language simulation program was integrated with an EIFFEL-based executive routine, which controls the simulation and I/O. The details of communication between EIFFEL modules and C-language modules was worked out. An EIFFEL module for graphical presentation of aircraft information was started. This was a graphics display of selected flight instruments, using X-Windows graphics. Every EIFFEL-based module is ultimately compiled into C-source, before final compilation into machine-dependent code.

The architecture shown in Figure-1, above, exemplifies a combination of what is called in computer science, *data-flow* modeling, and *object-oriented* modeling. ^[11] The data-flow architecture of computer science is the intuitive architecture of signal processing and control. However, object-orientation is invoked as a means to manage complexity in software development. Thus, the marriage of data-flow and object-orientation.

Next, a second-level architecture was created for the inference and control portions of the knowledge-based processor. This architecture is shown in Figure-2, below.

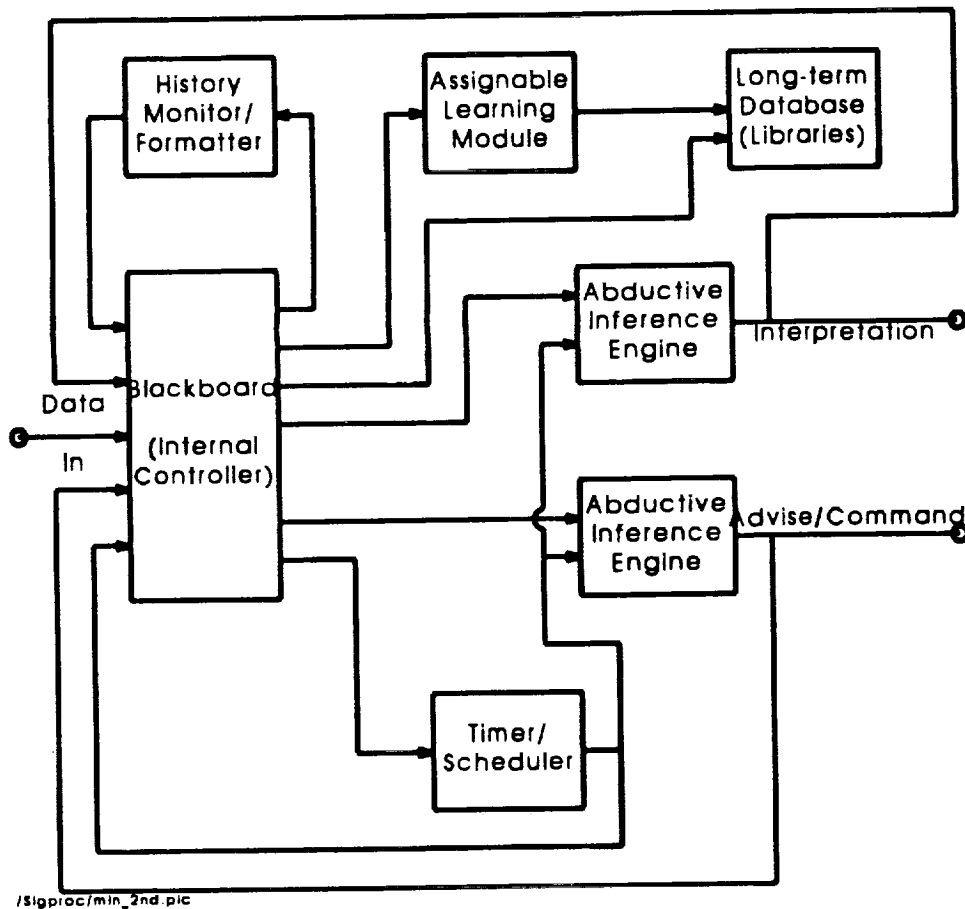


Figure 2. Symbolic Processor.

In the figure, the Interpreter is shown as an Abductive Inference Engine. The name derives from the fact that the inference implemented by the Interpreter is decision, and is a modification of Bayes or Fuzzy Decision to include "abductive inference." [12]

With respect to the Knowledge Processor of Figure-2., it should be noted that it produces two fundamentally different kinds of output. First, is the interpretation of what the vehicle is doing and how well it is doing it. Second, it produces autopilot commands and pilot advice. The Interpretation is the result of an inductive inference process. The Meta-Control is the result of a deductive

inference process. If these two symbolic inference processors were implemented using only rules, they would correspond to "forward-chaining" and "backward-chaining," respectively. However, our inference engines employ knowledge, represented by more than just sets of rules.

Other modules were added to the Knowledge Processor, as adjuncts to the basic tasks of Interpretation and Meta-Control. One was a Timer/Scheduler. This module is used, for instance, in making navigation calculations upon which autopilot inputs by the Meta-Controller are predicated. Also, a History Monitor/Formatter is used to support the Interpreter decisions. This formatter provides information which is used in a manner dual to that of correlation processing in Wiener or Kalman Filtering ^[13] A data handler, known as a Blackboard ^[14], is implemented in order to provide a standard set of data interfaces for intercommunication between the various software modules and communication with the outside world. This data handler greatly simplifies the problem of modifying and maintaining the individual modules during their evolution. Finally, a Neural module is indicated for future use, such as learning patterns for particular pilots, operations, or airports.

During the second year, the C-language simulation program was completed for the multi-engined jet transport. Autopilot flight control functions of Flight-Path Angle Hold and Airspeed Hold were incorporated, to provide a simulation interface suitable for the Meta-Controller or for external human control (Outer Loop). Work was started on the Graphical User Interface, Blackboard, Interpreter, and Meta-Controller. Each of these modules was assigned as an MS-Thesis or PhD-Dissertation project to four separate graduate students. At the end of the second year, preliminary versions of the four modules were being integrated into a running software system.

Effort began to represent the aircraft operating modes as discrete events (a partition) on the space of numerical state variables measurable from the simulation. Flight procedures, taken from the (American Airlines) Boeing-737 flight procedures manual, are being used to define the flight modes. Fuzzy Set Membership Functions are then defined to link sensor measurements (such as IAS, ALT, ROC, Gear, Flaps, EPR, etc.) to the flight modes.

The Principal Investigator visited the American Airlines Flight Simulation facility in Fort Worth, TX., and was given a two-hour ride in the full motion-base simulator for the Boeing-727. American Airlines is providing support at no cost to the present research effort. Later, Miss Glass, the developer of the Meta-Controller was taken as an observer for Line Orientation Flight Training in the Boeing-767 simulator, which possesses a state-of-the-art Flight Management System.

PLANS FOR THE THIRD YEAR:

It is planned to press on with the development, integration, and completion of the four principal modules of the Knowledge-Processor, being Blackboard, Graphical User Interface, Interpreter, and Meta-Controller. At present, there are

no visible impediments to completing on schedule.

As soon as practicable, a video is to be made of the project, concentrating on video of the Workstation screen (GUI), with running narration, during a simulated descent and approach to landing. The purpose of this video is to have a portable demonstration of the Knowledge-based Processor, in simulated operation. It is anticipated that the video would be attempted during the late Fall of 1991.

It is anticipated that several technical papers for submission to journals will be prepared during the winter of 1991-92. These will be prepared by the graduate students, paralleling their MS-Theses and/or PhD-Dissertation. Also, the Principal Investigator will prepare one paper, covering the entire project, with students as co-authors. These various journal submissions will also be distilled into a NASA Contractor Report.

The final task under the present three-year grant will be to reflect on what has been accomplished and how it has been accomplished, and to define the next step in this research area. The present grant represents a first pass through the very complex problem of Intelligent Control of Aircraft. The strategy has been to define and expose individual research problems and to obtain prototypical results. The results of this grant define the elements of the problem and investigate each element just deeply enough to get a prototype solution. The next step is to choose one or more of the individual elements for further and more comprehensive research and development. Continuing formulation of an underlying theoretical base is desired.

BIBLIOGRAPHY

1. Roskam, Jan; **Airplane Flight Dynamics and Automatic Flight Controls, Parts 1 and 2**, Roskam Aviation and Engineering Corp., Ottawa, Kansas, 1979.
2. Meyer, B.; **Object-Oriented Software Construction**; Prentice-Hall; 1988.
3. Sutherland, John W.; "Assessing the Artificial Intelligence Contribution to Decision Technology;" **IEEE Trans. Syst., Man, and Cyb.**; vol. SMC-16, #1, Jan/Feb. 1986; pp. 3-20.
4. Widman, L.E., Loparo, K.A., and Nielsen, N.R.; **Artificial Intelligence, Simulation, & Modeling**; Wiley; 1989.
5. Giarratano, J. and Riley, G.; **Expert Systems, Principles and Programming**; PWS-KENT Publ. Co., Boston; 1989.
6. Chandrasekaran, B.; "Generic Tasks in Knowledge-based Reasoning: High-level Building Blocks for Expert System Design;" **IEEE Expert**; Fall 1986, pp. 23-30.
7. Kuipers, Benjamin; "Qualitative Reasoning: Modeling and Simulation with Incomplete Knowledge;" **Automatica**; vol. 25, #4, 1989; pp. 571-585.

8. Pedrycz, W.; **Fuzzy Control and Fuzzy Systems**; Wiley; 1989.
9. Ramadge, P.J.G. and Wonham, W.M.; "The Control of Discrete Event Systems;" **Proceedings of the IEEE**; vol. 77, #1, Jan. 1989; pp. 81-98.
10. Schweppe, F.C.; **Uncertain Dynamic Systems**; Prentice-Hall; 1973; pp. 479-489.
11. Ward, Paul T., "How to Integrate Object Orientation With Structured Analysis and Design;" **IEEE Software**, March, 1989, pp. 74-82.
12. Punch, W.F., Tanner, M.C., Josephson, J.R., and Smith, J.W.; "Peirce, A Tool fo Experimenting With Abduction;" **IEEE Expert**; Oct. 1990, pp. 34-44.
13. Kalman, R.E.; "A New Approach to Linear Filtering and Prediction Problems;" **Trans. ASME, Journal Basic Eng.**; vol. 82D, Mar. 1960, pp. 34-45.
14. Nii, H.P.; "Blackboard Systems: The Blackboard Model of Problem Solving and the Evolution of Blackboard Architectures;" **The AI Magazine**; Summer, 1986; pp. 38-53.